

Turbo-Equalization Using Partial Gaussian Approximation

Chuanzong Zhang, Zhongyong Wang, Carles Navarro Manchón, Peng Sun, Qinghua Guo
and Bernard Henri Fleury, *Senior member, IEEE*

Abstract—This paper deals with turbo-equalization for coded data transmission over intersymbol interference (ISI) channels. We propose a message-passing algorithm that uses the expectation-propagation rule to convert messages passed from the demodulator-decoder to the equalizer and computes messages returned by the equalizer by using a partial Gaussian approximation (PGA). Results from Monte Carlo simulations show that this approach leads to a significant performance improvement compared to state-of-the-art turbo-equalizers and allows for trading performance with complexity. We exploit the specific structure of the ISI channel model to significantly reduce the complexity of the PGA compared to that considered in the initial paper proposing the method.

Index Terms—Turbo equalization, partial Gaussian approximation, message-passing.

I. INTRODUCTION

HISTORICALLY, turbo equalization of coded data transmission across a known inter-symbol interference (ISI) channel found its inspiration from turbo-decoding of turbo-codes, see [1] and references therein. Since its introduction turbo equalization has prevailed over more traditional equalization techniques available at that time due to its tremendous performance gain. Turbo-equalization is a collective name for joint data decoding and channel equalization algorithms that pass messages iteratively along the edges of a factor graph representing the probabilistic model of the considered transmission system. The most prominent message-passing algorithm – inherited from turbo-decoding of turbo-codes – is the sum-product algorithm [2], which is also known as belief propagation (BP) [3].

Two different factor graphs [2] representing the ISI channel can be drawn, which lead to different message-passing algorithms for equalization, see [4] and [5] for more details. In this letter, we use the one that exhibits a tree structure

[4]. This factor graph explicitly represents the channel state evolution, see Fig. 1. Applying BP on this graph yields BCJR-like equalization algorithms [4]. The complexity of these algorithms scales exponentially with the modulation order and the channel memory. Proposed solutions that circumvent this complexity problem convert the discrete messages returned by the demodulator-decoder into Gaussian functions that are passed as messages to the equalizer [6]–[8]. The complexity reduction results from the fact that the equalizer then processes Gaussian messages. The conversion can be done in two ways: either directly by matching the first and second moments of any of these discrete messages [7], or indirectly by using the formal rule of expectation propagation (EP) [9]: first a Gaussian approximation matching the first and second moments of the belief of the channel symbol node is computed from which the Gaussian message is then obtained [6], [8]. Numerical studies have shown that the latter conversion leads to better BER performance [6], [8].

Inspired by the partial Gaussian approximation (PGA) proposed in [10] we modify the messages returned from the equalizer and passed to the demodulator-decoder in [8]. The messages returned by the equalizer in [8] are computed from the above Gaussian-converted messages from the demodulator-decoder. By contrast, to equalize one channel symbol the new equalizer combines discrete messages from the demodulator-decoder for the symbols strongly interfering with said symbol and Gaussian-converted messages for the weakly interfering symbols. The reported simulation results show that doing so leads to a significant performance improvement compared to the turbo-equalizers in [7], [10] and [8]. Finally, our turbo-equalizer allows for trading complexity with performance by varying the set of symbols that are considered as strong interferers.

Our turbo-equalizer differs from the PGA-based one in [10] in two respects. First, in the former the conversion of the discrete messages returned by the demodulator-decoder into Gaussian functions is done using the formal EP rule, while it is performed by direct conversion of the discrete messages in the latter. Secondly, due to the particular structure of the ISI channel model, the messages returned by our equalizer can be computed from the Gaussian messages passed to it in a simple way. This leads to a significant complexity reduction compared to the turbo-equalizer in [10].

Notation– For a natural number N , we write $[N] = \{1, \dots, N\}$. Boldface lowercase and uppercase letters denote vectors and matrices, respectively. The identity matrix of size M is represented by \mathbf{I}_M . Superscript $(\cdot)^T$ designates

This work is supported by the National Natural Science Foundation of China (NSFC 61571402, NSFC U1204607, NSFC 61201251).

C. Zhang is with the School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China, and the Department of Electronic Systems, Aalborg University, Aalborg 9220, Denmark (e-mail: ieczhang@gmail.com).

Z. Wang and P. Sun are with the School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: iezy-wang@zzu.edu.cn; iepengsun@gmail.com).

C. Navarro Manchón and B. H. Fleury are with the Department of Electronic Systems, Aalborg University, Aalborg 9220, Denmark (e-mail: cnm@es.aau.dk; fleury@es.aau.dk).

Q. Guo is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia, and also with the School of Electrical, Electronic and Computer Engineering, the University of Western Australia, Crawley, WA 6009, Australia (e-mail: qguo@uow.edu.au).

transposition of a vector or matrix. We write $N(\mathbf{x}; \mathbf{m}, \mathbf{V})$ for the pdf of a multivariate Gaussian distribution with mean vector \mathbf{m} and covariance matrix \mathbf{V} . Depending on the context $\delta(\cdot)$ denotes either the Dirac delta function or the Kronecker delta. The relation $f(\cdot) = cg(\cdot)$ for some positive constant c is written as $f(\cdot) \propto g(\cdot)$. The notations $\sum_{\mathbf{x} \setminus \mathbf{y}} f(\mathbf{x})$ and $\int f(\mathbf{x}) d(\mathbf{x} \setminus \mathbf{y})$ denote respectively the partial summation and partial integration of the function $f(\mathbf{x})$ with respect to all entries of the vector \mathbf{x} except those entries common to \mathbf{x} and \mathbf{y} .

II. SYSTEM MODEL

The vector $\mathbf{b} = [b_1, \dots, b_K]^T$ of information bits is encoded and interleaved, yielding the codeword $\mathbf{c} = [c_1^T, \dots, c_N^T]^T$ with $\mathbf{c}_i = [c_i^1, \dots, c_i^Q]^T$. The coded bits are then mapped onto a Q -order modulation alphabet $\mathcal{X} \subseteq \mathbb{R}^1$, resulting in the vector of symbols $\mathbf{x} = [x_1, \dots, x_N]^T \in \mathcal{X}^N$. These symbols are transmitted over a linear, time-invariant, frequency-selective channel corrupted by additive white Gaussian noise (AWGN). The received vector $\mathbf{r} = [r_1, \dots, r_{N+L-1}]^T$ has entries

$$r_i = \sum_{l=0}^{L-1} h_l x_{i-l} + n_i = \mathbf{h}^T \mathbf{s}_i + n_i, \quad i \in [N+L-1] \quad (1)$$

where $\mathbf{s}_i = [x_{i-L+1}, \dots, x_i]^T$ with $x_i = 0$ for $i < 1$ and $i > N$, $\mathbf{h} = [h_{L-1}, \dots, h_0]^T$ represents the channel impulse response, and $\mathbf{n} = [n_1, \dots, n_{N+L-1}]^T$ is a white noise vector with component variance σ^2 .

A. Probabilistic Model and Factor Graph

The posterior probability mass function (pmf) of vectors \mathbf{b} , \mathbf{c} , \mathbf{x} and \mathbf{s} given the received signal \mathbf{r} reads

$$\begin{aligned} p(\mathbf{b}, \mathbf{c}, \mathbf{x}, \mathbf{s} | \mathbf{r}) &\propto \prod_{k=1}^K f_{B_k}(b_k) \times f_C(\mathbf{c}, \mathbf{b}) \\ &\times \prod_{i=1}^N f_{O_i}(r_i, \mathbf{s}_i) f_{T_i}(\mathbf{s}_i, \mathbf{s}_{i-1}, x_i) f_{M_i}(x_i, \mathbf{c}_i) \\ &\times \prod_{i=N+1}^{N+L-1} f_{O_i}(r_i, \mathbf{s}_i) f_{T_i}(\mathbf{s}_i, \mathbf{s}_{i-1}, 0) \end{aligned} \quad (2)$$

where $f_{B_k}(b_k)$ is the uniform prior pmf of the k th information bit, $f_C(\mathbf{c}, \mathbf{b})$ stands for the coding and interleaving constraints, $f_{O_i}(r_i, \mathbf{s}_i) \triangleq p(r_i | \mathbf{s}_i) = N(r_i; \mathbf{h}^T \mathbf{s}_i, \sigma^2)$ denotes the likelihood of \mathbf{s}_i , and $f_{M_i}(x_i, \mathbf{c}_i)$ represents the modulation mapping. Finally, $f_{T_i}(\mathbf{s}_i, \mathbf{s}_{i-1}, x_i)$ expresses the deterministic relationship between \mathbf{s}_i , \mathbf{s}_{i-1} and x_i , i.e.,

$$f_{T_i}(\mathbf{s}_i, \mathbf{s}_{i-1}, x_i) = \delta(\mathbf{G}\mathbf{s}_{i-1} + \mathbf{e}x_i - \mathbf{s}_i) \quad (3)$$

with the $L \times L$ matrix $\mathbf{G} = [\mathbf{0} \quad \mathbf{I}_{L-1}; \mathbf{0} \quad \mathbf{0}^T]$, $\mathbf{e} = [\mathbf{0}; 1]$ and $\mathbf{0}$ being a zero column vector of length $L-1$.

Fig. 1 depicts the factor graph [11] representing the factorization of the posterior pmf in (2). The factorization and its graph will be used as the baseline for the derivation

¹For simplicity we consider a real baseband model. The extension to a complex model is straightforward.

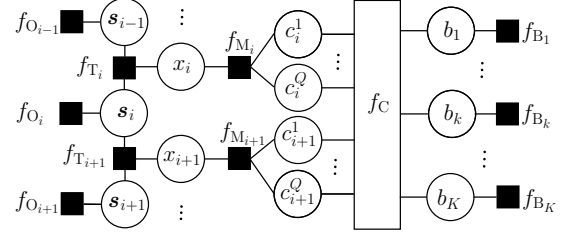


Fig. 1. Factor graph representing the probabilistic model (2).

of the turbo-equalizer described in Section III. To ease the subsequent discussions we identify two subgraphs. The channel subgraph includes the nodes of the channel symbols x_i , $i \in [N]$ and all factor nodes, variable nodes and edges “to the left” of these symbol nodes. The transmitter subgraph includes the channel symbol nodes and all factor nodes, variable nodes, and edges “to the right” of these symbol nodes.

III. DESIGN OF THE ITERATIVE RECEIVER

In a nutshell, we obtain the new turbo-equalizer by replacing the messages passed from the equalizer to the demodulator-decoder, i.e. from nodes f_{T_i} to nodes x_i , $i \in [N]$, in the turbo-equalizer in [8] by messages computed using the PGA approach in [10]. The next two subsections describe the messages computed in the new turbo-equalizer. The last subsection sketches the scheduling of these messages.

A. Equalization and Demodulation-decoding

Equalization and demodulation-decoding are implemented by passing messages along the edges of the channel subgraph and the transmitter subgraph respectively. Unless otherwise stated, these messages are computed using the BP rule [3].

1) *Demodulation-decoding*: The variables in the transmitter subgraph are discrete and so are the computed messages and beliefs. Decoding of the convolution code is done using the BCJR algorithm, an instance of BP. The messages from the modulator nodes to the channel symbol nodes are of the form $m_{f_{M_i} \rightarrow x_i}(x_i) \propto \sum_{\mathbf{x} \in \mathcal{X}} \beta_{\mathbf{x}} \delta(x_i - \mathbf{x})$ with $\beta_{\mathbf{x}} \geq 0$, $x_i \in \mathcal{X}$, $i \in [N]$.

2) *Equalization*: The latent variables \mathbf{s}_i , $i \in [N]$ in the channel subgraph are approximated as Gaussian variables. Since the channel is linear and noise is additive and Gaussian, the messages and beliefs are Gaussian functions. We write for the belief of node \mathbf{s}_i ($i \in [N]$),

$$b^G(\mathbf{s}_i) \propto N(\mathbf{s}_i; \mathbf{m}_{\mathbf{s}_i}, \mathbf{V}_{\mathbf{s}_i}). \quad (4)$$

The computation of this belief is given in [7] and [8, Eq. (28)].

B. Messages Exchanged Between the Equalizer and the Demodulator-decoder

Demodulator-decoder (D) \rightarrow Equalizer (E): The EP rule [9] is used to convert the discrete messages $m_{f_{M_i} \rightarrow x_i}(x_i)$, $i \in [N]$ into Gaussian messages [6], [8, Eq. (29)]:

$$\begin{aligned} m_{f_{M_i} \rightarrow x_i}^G(x_i) &= \frac{\text{Proj}_G[m_{f_{M_i} \rightarrow x_i}(x_i) n_{x_i \rightarrow f_{M_i}}^G(x_i)]}{n_{x_i \rightarrow f_{M_i}}^G(x_i)} \\ &\propto N(x_i; m_{x_i}, v_{x_i}), \quad i \in [N]. \end{aligned} \quad (5)$$

For a pdf $b(z)$, $\text{Proj}_{\mathcal{G}}[b(z)] = \arg \min_{b'(z) \in \mathcal{G}} D(b(z) || b'(z))$, with $D(\cdot || \cdot)$ denoting the Kullback-Leibler divergence and \mathcal{G} being the family of Gaussian pdfs. The parameters m_{x_i} and v_{x_i} in (5) are given by [8, Eq. (10) & (11)]. With this conversion, Gaussian messages $n_{x_i \rightarrow f_{T_i}}^G(x_i) = m_{f_{M_i} \rightarrow x_i}^G(x_i)$, $i \in [N]$ are passed to the equalizer.

$E \rightarrow D$: This is where the new turbo-equalizer differs from the one described in [6], [8].

In [6], [8] the Gaussian messages from f_{T_i} to x_i , $i \in [N]$ are converted into discrete messages²

$$m_{f_{T_i} \rightarrow x_i}(x_i) \propto m_{f_{T_i} \rightarrow x_i}^G(x_i), \quad x_i \in \mathcal{X}, i \in [N]. \quad (6)$$

The discrete messages $n_{x_i \rightarrow f_{M_i}}(x_i) = m_{f_{T_i} \rightarrow x_i}(x_i)$, $i \in [N]$ are then passed to the demodulator-decoder.

Consider a specific symbol x_i ($i \in [N]$). Clearly the computation of $m_{f_{T_i} \rightarrow x_i}(x_i)$ using (6) makes use of the Gaussian approximation of the messages from the other symbols, i.e. $n_{x_j \rightarrow f_{T_j}}^G(x_j)$, $j \in [N] \setminus \{i\}$ by the conversion (5). The idea is to use the original discrete messages rather than their Gaussian approximation for a selected subset of channel symbols which significantly interfere with x_i . It is inspired from the PGA proposed in [10].

First we identify those channel symbols “significantly” interfering with symbol x_i . Let $q_k = \sum_{l=0}^{L-1} h_l h_{l+k}$, $k \in \mathbb{Z}$ with $h_l = 0$ whenever $l \in \mathbb{Z} \setminus \{0, \dots, L-1\}$ be the autocorrelation function of the channel impulse response. Define $\mathbb{K}_\rho = \{k \in \{-(L-1), \dots, L-1\} : |q_k| > \rho q_0\}$ the set of lag indices at which the magnitude of the autocorrelation function is larger than ρq_0 , $\rho \in [0, 1)$. Then $\mathbb{I}_i^D = \{i+k : k \in \mathbb{K}_\rho\} \subseteq \mathbb{I}_i = \{i-(L-1), \dots, i+L-1\}$ contains the indices of the modulation symbol x_i and those symbols that interfere with x_i at correlation level ρ . We collect these symbols in the M -dimensional vector $\mathbf{x}_i^D = [x_j : j \in \mathbb{I}_i^D]^T$, with $M = |\mathbb{K}_\rho|$. We assume that $\bar{k} = \max \mathbb{K}_\rho$ fulfills $1 + 2\bar{k} \leq L$. Then we can readily show that all entries in \mathbf{x}_i^D are components of $\mathbf{s}_{i'}$ whenever $i + \bar{k} \leq i' \leq i + (L-1) - \bar{k}$. Notice that the assumption on \bar{k} guarantees that $i + \bar{k} \leq i + (L-1) - \bar{k}$.

With the above definitions we can now specify the message from f_{T_i} to x_i :

$$m_{f_{T_i} \rightarrow x_i}^{\text{PG}}(x_i) = \sum_{\kappa \in \mathbb{I}_i^D \setminus i} \frac{\prod_{\kappa \in \mathbb{I}_i^D \setminus i} n_{x_\kappa \rightarrow f_{T_\kappa}}(x_\kappa)}{\prod_{k \in \mathbb{I}_i^D} n_{x_k \rightarrow f_{T_k}}^G(x_k)} b_{i'}^G(\mathbf{x}_i^D) \quad (8)$$

where $n_{x_\kappa \rightarrow f_{T_\kappa}}(x_\kappa) = m_{f_{M_\kappa} \rightarrow x_\kappa}(x_\kappa)$ and $b_{i'}^G(\mathbf{x}_i^D) = \int b^G(\mathbf{s}_{i'}) d(\mathbf{s}_{i'} \setminus \mathbf{x}_i^D)$ with $b^G(\mathbf{s}_{i'})$ given in (4). The latter term is the belief of \mathbf{x}_i^D obtained by marginalization of the belief $b^G(\mathbf{s}_{i'})$. The index i' in $b_{i'}^G(\mathbf{x}_i^D)$ indicates that this belief depends on the time instant i' , $i + \bar{k} \leq i' \leq i + (L-1) - \bar{k}$. Notice that the selection $i' = i + \bar{k}$ minimizes the time instant ahead of i to wait for computing $m_{f_{T_i} \rightarrow x_i}(x_i)$. The derivation of (8) is provided in the appendix.

All Gaussian functions occurring in (8) combine as

$$\left[\prod_{\kappa \in \mathbb{I}_i^D} n_{x_\kappa \rightarrow f_{T_\kappa}}^G(x_\kappa) \right]^{-1} b_{i'}^G(\mathbf{x}_i^D) \propto N(\mathbf{x}_i^D; \mathbf{m}_{\mathbf{x}_i^D}^e, \mathbf{V}_{\mathbf{x}_i^D}^e) \quad (9)$$

²Strictly speaking, the message $m_{f_{T_i} \rightarrow x_i}(x_i)$ in (6) is the restriction of $m_{f_{T_i} \rightarrow x_i}^G(x_i)$ to \mathcal{X} .

with

$$\mathbf{V}_{\mathbf{x}_i^D}^e = \left[(\mathbf{P}_{i'} \mathbf{V}_{\mathbf{s}_{i'}} \mathbf{P}_{i'}^T)^{-1} - (\mathbf{V}_{\mathbf{x}_i^D}^e)^{-1} \right]^{-1}$$

$$\mathbf{m}_{\mathbf{x}_i^D}^e = \mathbf{V}_{\mathbf{x}_i^D}^e \left[(\mathbf{P}_{i'} \mathbf{V}_{\mathbf{s}_{i'}} \mathbf{P}_{i'}^T)^{-1} \mathbf{P}_{i'} \mathbf{m}_{\mathbf{s}_{i'}} - (\mathbf{V}_{\mathbf{x}_i^D}^e)^{-1} \mathbf{m}_{\mathbf{x}_i^D} \right].$$

In these expressions, the $M \times L$ selection matrix $\mathbf{P}_{i'}$ extracts the vector \mathbf{x}_i^D from $\mathbf{s}_{i'}$, i.e. $\mathbf{x}_i^D = \mathbf{P}_{i'} \mathbf{s}_{i'}$. The entries of the vector $\mathbf{m}_{\mathbf{x}_i^D}$ and the diagonal entries of the diagonal matrix $\mathbf{V}_{\mathbf{x}_i^D}$ are the first moments m_{x_κ} and the second central moments v_{x_κ} respectively of the messages $n_{x_\kappa \rightarrow f_{T_\kappa}}^G(x_\kappa)$, $\kappa \in \mathbb{I}_i^D$. Inserting (9) into (8) yields the PGA-based messages

$$m_{f_{T_i} \rightarrow x_i}^{\text{PG}}(x_i) = \sum_{\mathbf{x}_i^D \setminus x_i} N(\mathbf{x}_i^D; \mathbf{m}_{\mathbf{x}_i^D}^e, \mathbf{V}_{\mathbf{x}_i^D}^e) \prod_{\kappa \in \mathbb{I}_i^D \setminus i} n_{x_\kappa \rightarrow f_{T_\kappa}}(x_\kappa), \quad i \in [N] \quad (10)$$

that replace the messages $m_{f_{T_i} \rightarrow x_i}^G(x_i)$, $i \in [N]$ in (6).

C. Messages Scheduling

The turbo-equalizer implements the following scheduling:

- S1: *Initialization*: $n_{x_i \rightarrow f_{T_i}}(x_i) \propto 1$ and $n_{x_i \rightarrow f_{T_i}}^G(x_i) = \mathcal{N}(x_i; 0, 1)$, $i \in [N]$.
- S2: *Equalization*: The messages $m_{f_{T_i} \rightarrow \mathbf{s}_i}^G(\mathbf{s}_i)$ and $n_{\mathbf{s}_i \rightarrow f_{T_{i+1}}}^G(\mathbf{s}_i)$, $i \in [N+L-1]$ are recursively computed using (12) and (15) respectively in [8]. In parallel, the messages $m_{f_{T_i} \rightarrow \mathbf{s}_{i-1}}^G(\mathbf{s}_{i-1})$ and $n_{\mathbf{s}_{i-1} \rightarrow f_{T_{i-1}}}^G(\mathbf{s}_{i-1})$, $i \in \{N+L-1, \dots, 1\}$ are recursively calculated from (18) and (21) respectively in [8]. Finally, the beliefs $b^G(\mathbf{s}_i)$, $i \in [N]$ (see (4)) are obtained by (28) in [8].
- S3: $E \rightarrow D$: The messages $m_{f_{T_i} \rightarrow x_i}^{\text{PG}}(x_i)$, $i \in [N]$ are obtained from (9) and (10).
- S4: *Demodulation-decoding*: The messages $m_{f_{T_i} \rightarrow x_i}^{\text{PG}}(x_i)$, $i \in [N]$ are passed to the demodulator. The BCJR algorithm, an instance of BP, is run in the decoder, yielding the discrete messages $n_{x_i \rightarrow f_{T_i}}(x_i) = m_{f_{M_i} \rightarrow x_i}(x_i)$, $i \in [N]$.
- S5: $D \rightarrow E$: The Gaussian messages $n_{x_i \rightarrow f_{T_i}}^G(x_i) = m_{f_{M_i} \rightarrow x_i}^G(x_i)$, $i \in [N]$ are updated using (5).

Steps S2–S5 constitute an iteration that is repeated until a maximum number of iterations is reached.

IV. ANALYSIS, PERFORMANCE AND COMPLEXITY

A. Comparison with Existing Turbo-equalizers

We compare the performance of the new turbo-equalizer (we denote it as BP-EP-PGA) with that of three other turbo-equalizers published in the literature by means of Monte Carlo simulations: (1) BP-EP: the combined BP-EP algorithm in [8]; (2) BP-PGA: an implementation of the PGA algorithm in [10] for the equalization of ISI channels; (3) BP-GA: the LMMSE-based turbo-equalizer, which is equivalent to Gaussian-approximated BP [7]. In our implementation BP-PGA is obtained from BP-EP-PGA by substituting the EP rule (5) with a direct Gaussian approximation of the discrete messages from f_{M_i} to x_i , $i \in [N]$. All four turbo-equalizers solely differ in the types of messages exchanged between the equalizer and the demodulator-decoder. The table below reports these distinctive features.

$$b^G(\mathbf{s}_{i'}) = m_{f_{T_{i'+1}} \rightarrow \mathbf{s}_{i'}}^G(\mathbf{s}_{i'}) m_{f_{O_{i'}} \rightarrow \mathbf{s}_{i'}}^G(\mathbf{s}_{i'}) \left[\prod_{l=0}^{L-1} n_{x_{i'-l} \rightarrow f_{T_{i'-l}}}^G(x_{i'-l}) \right] \\ \times \int \left[\prod_{l=1}^{L-1} m_{f_{O_{i'-l}} \rightarrow \mathbf{s}_{i'-l}}^G(\mathbf{s}_{i'-l}) \right] n_{\mathbf{s}_{i'-L} \rightarrow f_{T_{i'-L+1}}}^G(\mathbf{s}_{i'-L}) d\mathbf{s}_{i'-L} \quad (7)$$

Turbo-equalizer	$D \rightarrow E$	$E \rightarrow D$
BP-GA [7]	Direct conversion	GA
BP-EP [8]	EP-rule	GA
BP-PGA [10]	Direct conversion	PGA
BP-EP-PGA (new)	EP-rule	PGA

As the selected threshold ρ in BP-EP-PGA approaches 1, \mathbb{I}_i^D typically shrinks to the singleton $\{i\}$ ($M = 1$). With this configuration, the messages $m_{f_{T_i} \rightarrow x_i}^{PG}$, $i \in [N]$ in (10) coincide with the messages $m_{f_{T_i} \rightarrow x_i}^G$, $i \in [N]$ in (6) and, consequently, BP-EP-PGA and BP-EP become equivalent. Notice that both schemes compute the same messages in stage S2-Equalization. They solely differ in S5- $D \rightarrow E$.

B. Computational Complexity

The complexity of the PGA algorithm in [10], which was designed for generic channel matrices, is $\mathcal{O}(N^2 + M^2 Q^M)$ per symbol. The main contribution to the complexity of the BP-EP-PGA is at (4), which requires $L \times L$ matrix inversions (see [8, Eq. (28)]), and at (8). Thus, the complexity is $\mathcal{O}(L^3 + M^2 Q^M)$ per symbol. The complexity reduction method described in [8, Subsec. IV.C] can, however, also be applied to BP-PGA and BP-EP-PGA. Since the beliefs $b_{i'}^G(\mathbf{x}_i^D)$ (see (8)) are obtained from the beliefs $b_{i'}^G(\mathbf{s}_{i'})$, $i \in [N]$ (4) needs only be computed once every $(L-M+1)$ symbols when $\mathbb{K}_\rho = \{-\bar{k}, \dots, \bar{k}\}$ ($M = 1 + 2\bar{k}$). In this case, the complexity of BP-PGA and BP-EP-PGA is $\mathcal{O}(L^3/(L-M+1) + M^2 Q^M)$ per symbol.

C. Numerical Assessment

We compare the BER performance of the four above turbo-equalizers and a receiver designed for and operating in a non-dispersive AWGN channel.

A sequence of 2048 information bits is encoded using a 1/2 rate convolutional code with generator polynomials $(23, 35)_8$. The coded bits are interleaved and then mapped onto BPSK symbols ($\mathcal{X} = \{-1, +1\}$), which are transmitted over a severely distorted ISI channel with impulse response $\mathbf{h} = [0.227 \ 0.460 \ 0.668 \ 0.460 \ 0.227]^T$. The BER performance is evaluated after 30 turbo-equalization iterations. For BP-PGA and BP-EP-PGA, we set ρ so that $M = 3$.

The results are depicted in Fig. 2. We observe a remarkable performance improvement of BP-EP-PGA compared to the other turbo-equalizers. We attribute this improvement to the fact that BP-EP-PGA combines the advantages from both BP-EP and BP-PGA. Firstly, implementing the EP-based conversion (5) instead of a direct conversion of the discrete messages from f_{M_i} to x_i , $i \in [N]$ provides an advantage over BP-GA and BP-PGA. Secondly, implementing (10) leads to better performance than when computing the right-hand messages in

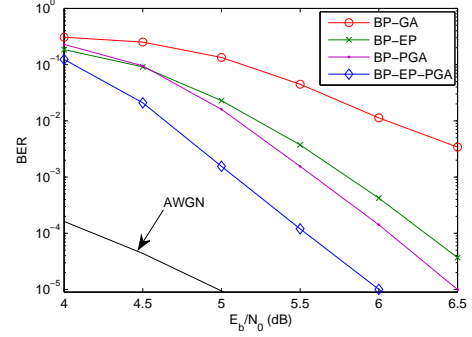


Fig. 2. BER performance of the considered turbo-equalizers.

(6) at the expense of a complexity increase, as can be seen by comparing BP-EP and BP-EP-PGA. Since BP-EP can be seen as an instance of our proposed BP-EP-PGA with the setting $M = 1$, we conclude that the tuning of the parameter M (or equivalently ρ) allows for trading performance and computational complexity in the receiver.

APPENDIX

We compute (8) for a specific symbol x_i ($i \in [N]$). Select $\mathbf{s}_{i'}$ with $i + \bar{k} \leq i' \leq i + (L-1) - \bar{k}$, see $E \rightarrow D$ in Subsection III-B. By applying the BP rule we obtain for the Gaussian belief of $\mathbf{s}_{i'}$

$$b^G(\mathbf{s}_{i'}) = m_{f_{T_{i'}} \rightarrow \mathbf{s}_{i'}}^G(\mathbf{s}_{i'}) m_{f_{O_{i'}} \rightarrow \mathbf{s}_{i'}}^G(\mathbf{s}_{i'}) m_{f_{T_{i'+1}} \rightarrow \mathbf{s}_{i'}}^G(\mathbf{s}_{i'}). \quad (11)$$

To compute $m_{f_{T_{i'}} \rightarrow \mathbf{s}_{i'}}^G(\mathbf{s}_{i'})$ we use the BP rule in a forward recursion along the variable and factor nodes $f_{T_{i'-L+1}}, \mathbf{s}_{i'-L+1}, \dots, \mathbf{s}_{i'-1}, f_{T_{i'}}$. Doing so and inserting in (11) yields the expression in (7). Notice that the product in the first pair of brackets and the integral are functions of $\mathbf{s}_i = [x_{i-L+1}, \dots, x_i]^T$. From the choice of i' , the entries of $\mathbf{x}_i^D = [x_j : j \in \mathbb{I}_i^D]^T$ are also entries of $\mathbf{s}_{i'}$, see $E \rightarrow D$ in Subsection III-B. Thus, the product in the first bracket in (7) contains as factors the messages $n_{x_j \rightarrow f_{T_{i'}}}^G(x_j)$, $j \in \mathbb{I}_i^D$. We implement a PGA by substituting these messages with their discrete counterparts $n_{x_j \rightarrow f_{T_{i'}}}(x_j)$, $j \in \mathbb{I}_i^D$. This substitution can be formally expressed as

$$b^{PG}(\mathbf{s}_{i'}) = \prod_{\kappa \in \mathbb{I}_i^D} \frac{n_{x_\kappa \rightarrow f_{T_{i'}}}(x_\kappa)}{n_{x_\kappa \rightarrow f_{T_{i'}}}^G(x_\kappa)} b^G(\mathbf{s}_{i'}). \quad (12)$$

By using the marginalization constraint of BP we can write

$$m_{f_{T_{i'}} \rightarrow x_i}^{PG}(x_i) n_{x_i \rightarrow f_{T_{i'}}}(x_i) = \sum_{\mathbf{x}_i^D \setminus x_i} \int b^{PG}(\mathbf{s}_{i'}) d(\mathbf{s}_{i'} \setminus \mathbf{x}_i^D). \quad (13)$$

Notice that the right-hand term is a marginal belief of x_i . Solving for $m_{f_{T_{i'}} \rightarrow x_i}^{PG}(x_i)$ in (13) yields (8).

REFERENCES

- [1] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, pp. 754 – 767, May 2002.
- [2] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Trans. Signal Processing*, pp. 28 – 40, Jan. 2004.
- [3] J. Pearl, "Reverend Bayes on inference engines: a distributed hierarchical approach," in *Proceedings of the National Conference on Artificial Intelligence*, 1982, pp. 133 – 136.
- [4] B. Kurkoski, P. Siegel, and J. Wolf, "Joint message-passing decoding of ldpc codes and partial-response channels," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1410 – 1422, Jun 2002.
- [5] G. Colavolpe and G. Germini, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 818 – 825, Apr. 2005.
- [6] J. Hu, H.-A. Loeliger, J. Dauwels, and F. Kschischang, "A general computation rule for lossy summaries/messages with examples from equalization," in *Proc. 44th Allerton Conf. Communication, Control, and Computing*, Sep. 2006, pp. 27 – 29.
- [7] Q. Guo and L. Ping, "LMMSE turbo equalization based on factor graphs," *IEEE J. Select. Areas Commun.*, vol. 26, no. 2, pp. 311 – 319, Feb. 2008.
- [8] P. Sun, C. Zhang, Z. Wang, C. N. Manchón, and B. H. Fleury, "Iterative receiver design for ISI channels using combined belief- and expectation-propagation," *IEEE Signal Processing Lett.*, vol. 22, no. 10, pp. 1733 – 1737, Oct. 2015.
- [9] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the Seventeenth Conf. on Uncertainty in Artificial Intelligence*, 2001, pp. 362 – 369.
- [10] Q. Guo, D. Huang, L. Ping, S. Nordholm, J. Xi, and P. Li, "Soft-in soft-out detection using partial Gaussian approximation," *IEEE Access*, vol. 2, pp. 427 – 436, May 2014.
- [11] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498 – 519, Feb. 2001.